INFORMATION AND COMPUTATION 97, 97–131 (1992)

An Algebraic Model of Synchronous Systems

MIKLÓS BARTHA*

Bolyai Institute, A. József University, 6720 Szeged, Hungary

Finite state structural Mealy automata over an algebraic theory T (called structural T-automata) are introduced to model behaviors of synchronous systems. The main result is a left adjoint construction which extends the algebraic theory T to a strong feedback theory $F_s T$ by adjoining the operation of feedback to it. Structural T-automata equipped with simulations as vertical arrows between them form a symmetric monoidal 2-category. $F_s T$ is obtained by divesting this 2-category of its vertical structure, i.e., by making equivalent all the automata contained in the same connected component of a given hom-category. It is shown that, up to isomorphism of 2-cells, each equivalence class contains a unique automaton which is minimal regarding the number of its registers.

Contents

1. INTRODUCTION

Systolic flowchart schemes and feedback theories were introduced in (Bartha, 1987b) to provide a new calculus for the study of systolic systems. In this paper we replace the attribute "systolic" by "synchronous", which is more adequate to the model we are going to deal with.

Synchronous flowchart schemes arise naturally from the combination of two models of computation, the connection of which was not recognized until recently. The motivation stems from (Leiserson, 1982; Leiserson and

* This paper was written during the author's visit at Oxford University Computing Laboratory, Oxford, U.K., and its revision took place at the Department of Computer Science, The University of Western Ontario, London, Ontario, Canada. The author's current address for correspondence is the Department of Computer Science, Memorial University of Newfoundland, St. John's, Newfoundland, Canada A1C 5S7.

0890-5401/92 $3.00

Saxe, 1983), where the reader can find a mainly technical description of synchronous systems. In (Leiserson and Saxe, 1983) the authors define the model of a synchronous system as a finite edge-weighted directed multigraph $G=(V, E)$, called a communication graph. The vertices V of this communication graph correspond to functional elements, and the edges E correspond to interconnections between the functional elements. Each edge e in E is a triple of the form (u, v, w), where u and v are (possibly identical) vertices of G, called the tail and the head of e, and w is the nonnegative integer weight of e. The weight specifies the delay (measured in clock cycles) by which data are transferred from the tail vertex to the head one. In other words, the weight is the number of registers placed along the interconnection between the functional elements. A configuration of the system is some assignment of values to all its registers. With each clock tick, the system maps the current configuration into a new one. If the weight of an edge is zero, then data ripple through that edge always in the same clock cycle. To avoid feedback of rippling, which would cause problems of latching or oscillation, it is required that in every cycle of G there be at least one edge which has strictly positive weight. In this model, one vertex called the host represents the external interface.

Observe that a synchronous system can be considered as a Mealy automaton whose states are represented by the collection of all registers, and the transition function is the collection of all functional elements interpreted in an appropriate way. The transition function in this case is also called the combinational logic. We adopt the terminology "finite state structural Mealy automaton" from (Gécseg and Peák, 1972) for the present analysis of such decomposable machines. Structural Mealy automata are in fact generalizations of linear systems that are well-known from control theory; see, e.g., (Padulo and Arbib, 1974; Kalman *et al.*, 1969; Arbib and Manes, 1974b). The meaning of "decomposable" in linear systems is, however, different from what it meant above.

Two critical notes must be made about the above model:

1. According to Leiserson (1982), a semisystolic system, which corresponds to a synchronous system here, is rather an *infinite* edge-weighted directed multigraph represented by its finite approximations that must be regular in a certain sense. This conception is underlined also in (Leiserson and Saxe, 1983) by the given example of a real time palindrome recognizer. Therefore the single finite graph G should be called a finite system only, or rather a *scheme*.

2. The multigraph representation of a synchronous scheme is inadequate in the sense that it does not relate the two endpoints of a given edge to designated labelled input–output "ports" of the corresponding vertices. This question is clearly important, because different i/o ports of the

functional elements (processors) behave differently in general. Also, it is advantageous to replace the host by a fixed (finite) number of *input–output channels* as distinguished vertices, thus avoiding the unnecessary constraint that those cycles closed up only by the host should also contain an edge with positive weight.

These two criticisms suggest reconsidering synchronous systems in the framework of Elgot's (1975) well-known model of monadic computations (flowchart algorithms). This standpoint motivated the definition of synchronous flowchart schemes in (Bartha, 1987b). The very same model, equipped with so called topological transformations instead of the algebraic structure, was defined already in (Culik and Fris, 1985). With our approach it becomes possible to study synchronous systems in an exact algebraic (and/or category theoretical) framework, adopting the sophisticated techniques and constructions developed for flowchart schemes and iteration theories.

To describe sychronous schemes and their behavior we use a many-sorted algebraic language (type) consisting of composition, sum, and feedback as basic operations, and only five constants: 0, 1, x, ε, 0_1. Sometimes we switch to a category theoretical interpretation of our algebras, in which case sum is a bifunctor making the underlying category strictly monoidal, see (MacLane, 1971). The constant x then provides a symmetry for the tensor sum. Category theory was first used by Hotz (1965, 1966) to discuss switching circuits. In the latter paper Hotz used "x" in "x-category" in much the same spirit as we do below. The operation of feedback ($\uparrow$) was first defined in (Ştefănescu, 1986) as a substitute for iteration in iteration theories. Independently, in (Bartha, 1987a), the same operation, together with a new treatment of the above constants, was introduced to provide a finite axiomatization of flowchart schemes. For a different axiomatization of flowchart schemes, see (Bloom and Ésik, 1985).

Synchronous (systolic) schemes were axiomatized in (Bartha, 1987b). It was pointed out there that from an axiomatic point of view the only difference between ordinary and sychronous schemes lies in the different treatment of the derived algebraic constant $\boldsymbol{\alpha} = \uparrow x$. (That paper used the symbol $\mathbf{V}$ to denote this constant.) Not surprisingly, this comparison can be extended to the axiomatization of the algebras describing the semantics of the two kinds of schemes, i.e., to iteration and feedback theories. Accordingly, the same parallel can be set up between our feedback theories and those defined in (Ştefănescu, 1986).

Iteration theories were axiomatized in (Ésik, 1980). In terms of axiomatization, strong iteration theories can be obtained from iteration theories by replacing one group of identities (the commutativity axioms, see below) by a stronger implication. This implication is— as observed by

Ésik (1988)—still satisfied in most of the known subclasses of iteration theories, namely in rational algebraic theories (Wright *et al.*, 1976), metric iteration theories (Troeger, 1982), pointed iterative theories (Bloom *et al.*, 1980a, 1980b), iteration theories fulfilling the functoriality axiom (Arbib and Manes, 1980) and flowchart theories (Ştefănescu, 1986, 1987; Căzănescu and Ştefănescu, 1988). Moreover, it is easy to see that the same implication is satisfied also in feedback theories obtained by the F^{∞}-construction (Bartha, 1989). In categorical terminology, this implication is known as the weak functorial dagger; see (Arbib and Manes, 1980). An axiomatization of strong iteration theories using the operations composition, sum, and feedback can be found in (Căzănescu and Ştefănescu, 1988). In Section 3 we introduce strong iteration (feedback) theories by a simpler axiom as opposed to the implication mentioned above.

This paper is organized as follows. Sections 2 and 3 give a short summary of the most important axiomatization results on feedback and iteration theories. In Section 4 we define generalized finite state structural Mealy automata, called structural T-automata, in which the combinational logic is a morphism in an arbitrary algebraic theory T. The main result (Sections 4 and 5) is a construction of the free strong feedback theory generated by T, denoted $F_s T$. The theory $F_s T$ is obtained as the quotient of the algebra of structural T-automata by a congruence relation which is analogous to the well-known equivalence of finite state automata arising from their minimization. It is only in Section 4 that we really apply category theory to discuss the basic properties of our automata in the most general framework of categorical machines by Arbib and Manes (1974a). The category of structural T-automata is enriched by vertical arrows describing simulations between the automata. It is shown that, up to isomorphism of 2-cells, the resulting structure is a symmetric monoidal 2-category. Splitting the vertical categories of this 2-category into their connected components, in each such component there exists an object (i.e., T-automaton) having the fewest registers. This minimal object is initial among those, contained in the same connected component, that are reduced in a certain natural sense. In Section 6 we construct a counterexample showing that the free strong iteration theory generated by T cannot be obtained from $F_s T$ in the expected obvious way. Finally, Section 7 contains an interesting note on systolic computer architectures.

2. Preliminaries

It was observed by Elgot and Shepherdson (1979) that flowchart schemes can be treated as morphisms in a strict monoidal category (MacLane, 1971) over the set of objects $N = \{0, 1, 2, ...\}$. Arnold and

Dauchet (1978, 1979) reformulated these categories as $N \times N$-sorted algebras called magmoids. In a magmoid M, we have an underlying set $M(p, q)$ corresponding to each pair (p, q) of nonnegative integers, and the basic operations are the following:

Composition: maps $M(p, q) \times M(q, r)$ into $M(p, r)$ for each triple p, q, r in N. Composition is denoted by $\cdot$, as usual.

Sum: maps $M(p_1, q_1) \times M(p_2, q_2)$ into $M(p_1 + p_2, q_1 + q_2)$ for each choice of the nonnegative integers p_1, p_2, q_1, q_2. Sum is denoted by $+$.

There are two constants in M, 0 and 1, standing for the identity arrows 1_0 and 1_1, respectively. By the strict monoidal property, 1_p $(p \geqslant 1)$ then corresponds to the element $\sum_{i=1}^{p} 1$ in $M(p, p)$. We use the (ambiguous) notation p for $\sum_{i=1}^{p} 1$, and adopt the categorical terminology $f: p \to q$ to mean that f is an element (morphism) of sort (p, q) in M. The operations and constants are subject to the obvious identities M1, ..., M5 below. These identities were earlier considered in (Benson, 1975).

The magmoid operations are, however, not sufficient to express even the most elementary schemes, i.e., mappings. For this reason, some further constants are to be introduced. Usually the (infinite number of) constants π_p^i for all $p \in N$ and $i \in [p] = \{1, 2, ..., p\}$ are chosen. The constant $\pi_p^i: 1 \to p$ represents the mapping $[1] \to [p]$ which sends 1 to i. This choice is natural, because the semantics of flowchart schemes is defined in algebraic theories (in the sense of Lawvere (1963)), and the constants π_p^i are included in the type of the corresponding $N \times N$-sorted algebras, see Elgot's (1975) "distinguished morphisms" i. However, regarding the pure syntax of schemes only, the choice of the constants π_p^i is not the simplest one. Indeed, every mapping can be expressed by the help of the transposition $x: 2 \to 2$, the join (or branch) $\varepsilon: 2 \to 1$, and the zero $0_1: 0 \to 1$ using the magmoid operations. These constants are also natural for us, even from the semantic point of view, because we consider schemes to be logical circuits. In this case the constants x, ε and 0_1 are interpreted as the simplest switching elements in the curcuits; see (Bartha, 1987b, Fig. 3). To deal with feedback in the algebra of schemes, we enrich the magmoid operations by the adequate unary operation

Feedback: maps $M(1+p, 1+q)$ into $M(p, q)$ for each pair $(p, q) \in N \times N$. Feedback is denoted by $\uparrow$.

In accordance with (Bartha, 1987b), S denotes the type consisting of the operations $\cdot$, $+$, and $\uparrow$ and constants 0, 1, x, ε, and 0_1, and D is the subtype of S not containing $\uparrow$. In addition, when dealing with algebraic theories (theories, for short) we often use tupling $(\langle \cdots \rangle)$ and the injections

π_p^i as derived operations, cf. (Bartha, 1987a, 1987b). For simplicity, the notation π^i is preferred to π_p^i if p is understood. Mappings, except for some distinguished ones, e.g., x, 0_1, w_p, will be denoted by lowercase Greek letters throughout the paper. For a mapping $\beta: [p] \to [q]$ we use the slightly ambiguous notation $\beta: p \to q$. The following mappings will play an important role in the sequel:

$\varepsilon_k: k \to 1$ is the unique mapping of its sort.

$w_p(q): p \cdot q \to q$. For any $p, q \in N$, $w_p(q)$ takes a number of the form $(j-1) \cdot q + i$ ($j \in [p]$, $i \in [q]$) to i. Note that $w_p(1) = \varepsilon_p$, and $w_0(q) = 0_q$ is the unique mapping $0 \to q$.

$\kappa(n, p): p \cdot n \to n \cdot p$. This permutation (sometimes called a perfect shuffle) rearranges p blocks of length n into n blocks of length p, i.e., $\kappa(n, p)$ takes $(j-1) \cdot n + i$ ($j \in [p]$, $i \in [n]$) to $(i-1) \cdot p + j$.

$\beta \# s$. If $\beta: r \to r$ is any permutation and s is a sequence $(n_1, ..., n_r)$ of nonnegative integers with $n = \sum_{i=1}^{r} n_i$, then $\beta \# s: n \to n$ is the block by block performance of β on s, i.e., $\beta \# s$ sends $j + \sum_{i=1}^{k} n_i$, where $j \in [n_{k+1}]$ to the number $y + j$, where y is the sum of numbers n_i such that $\beta(i) < \beta(k+1)$.

If Q is any type of $N \times N$-sorted algebras and E is a set of Q-identities, then we denote by $\mathscr{K}_Q(E)$ the variety of all Q-algebras in which the identities E are valid. If A is a Q-algebra, then $\Omega_A(E)$, or simply $\Omega(E)$, denotes the congruence relation of A induced by E, i.e., the smallest congruence relation for which $A/\Omega(E)$ (the quotient of A by $\Omega(E)$) becomes an algebra in $\mathscr{K}_Q(E)$.

In (Bartha, 1987a, 1987b, 1988) two systems of identities FT and IT were developed systematically to serve as bases of identities of feedback and iteration theories, respectively. FT and IT can be constructed through the following steps.

1. $\mathrm{MG} = \{\mathrm{M1}, ..., \mathrm{M5}\}$ is the set of magmoid identities, (cf. Arnold and Dauchet, 1978) where

M1: $f \cdot (g \cdot h) = (f \cdot g) \cdot h$ if $f: p \to q$, $g: q \to r$, $h: r \to s$;

M2: $f + (g + h) = (f + g) + h$ if $f: p_1 \to q_1$, $g: p_2 \to q_2$, $h: p_3 \to q_3$;

M3: $p \cdot f = f \cdot q = f$ if $f: p \to q$;

M4: $f + 0 = 0 + f = f$ if $f: p \to q$;

M5: $(f_1 \cdot g_1) + (f_2 \cdot g_2) = (f_1 + f_2) \cdot (g_1 + g_2)$ if $f_i: p_i \to q_i$, $g_i: q_i \to r_i$, $i = 1, 2$.

2. $\mathrm{DF} = \mathrm{MG} \cup \{\mathrm{P}, \mathrm{D1}, \mathrm{D2}, \mathrm{D3}\}$, where

P: $f_1 + f_2 = x \# (p_1, p_2) \cdot (f_2 + f_1) \cdot x \# (q_2, q_1)$ if $f_i: p_i \to q_i$, $i = 1, 2$.

P is the block permutation axiom introduced by Elgot and Shepherdson

(1980). This axiom postulates a *symmetry* (McLane, 1971) for the strict monoidal category determined by the axioms MG.

$D1$: $(\varepsilon+1)\cdot\varepsilon=(1+\varepsilon)\cdot\varepsilon$;

D2: $x\cdot\varepsilon=\varepsilon$;

D3: $(1+0_1)\cdot\varepsilon=1$.

3. SF = DF $\cup$ {S1, S2, ..., S9} and
4. SC = DF $\cup$ {S1, S2, ..., S6, X}, where

S1: $\uparrow(f_1+f_2)=\uparrow f_1+f_2$ if $f_1\colon 1+p_1\to 1+q_1, f_2\colon p_2\to q_2$;

S2: $\uparrow^2((x+p)\cdot f)=\uparrow^2(f\cdot(x+q))$ if $f\colon 2+p\to 2+q$;

S3: $\uparrow(f\cdot(1+g))=(\uparrow f)\cdot g$ if $f\colon 1+p\to 1+q$, $g\colon q\to r$;

S4: $\uparrow((1+g)\cdot f)=g\cdot\uparrow f$ if $f\colon 1+q\to 1+r$, $g\colon p\to q$;

S5: $\uparrow 1=0$;

S6: $\varepsilon\cdot\perp=\perp+\perp$, where $\perp=\uparrow\varepsilon$;

S7: $\uparrow(f\cdot(\varepsilon+q))=\uparrow^2((\varepsilon+p)\cdot f)$ if $f\colon 1+p\to 2+q$;

S8: $0_1\cdot\boldsymbol{\alpha}=0_1$, where $\boldsymbol{\alpha}=\uparrow x$;

S9: $\uparrow(\varepsilon\cdot\boldsymbol{\alpha}^n)=\perp$ for all $n\in N$, where $\boldsymbol{\alpha}^n$ denotes the n-fold composite of $\boldsymbol{\alpha}$;

X: $\boldsymbol{\alpha}=1$.

5. FT = SF $\cup$ {TH, C} and IT = SC $\cup$ {TH, C}, where

TH:

$$w_p(p)\cdot f=\left(\sum_{i=1}^{p} f\right)\cdot w_p(q) \qquad \text{if} \quad f\colon p\to q,$$

see also (Bartha, 1987a, 1987b), and

C:

$$w_n(p)\cdot\uparrow^l f=\uparrow^{l\cdot n}(f*(\rho_1,...,\rho_l))\cdot w_n(q) \qquad \text{if} \quad f\colon l+p\to l+q,$$

for all $n\in N$ under every choice of the mappings $\rho_1,...,\rho_l\colon n\to n$, where

$$f*(\rho_1,...,\rho_l)=\alpha(l,n,p)^{-1}\cdot\left(\sum_{i=1}^{n} f\right)\cdot\alpha(l,n,q)\cdot\left(\sum_{i=1}^{l}\rho_i+n\cdot q\right)$$

and $\alpha(l,n,m)=(\kappa(2,n)\#(l,m)^n)\cdot(\kappa(l,n)+n\cdot m)$, see also Fig. 1.

The identity TH is called the theory identity for a reason explained below. C is the group of commutativity axioms introduced by Ésik (1980) in a different form. The reader is referred to (Bartha, 1988) for a more detailed explanation of C.

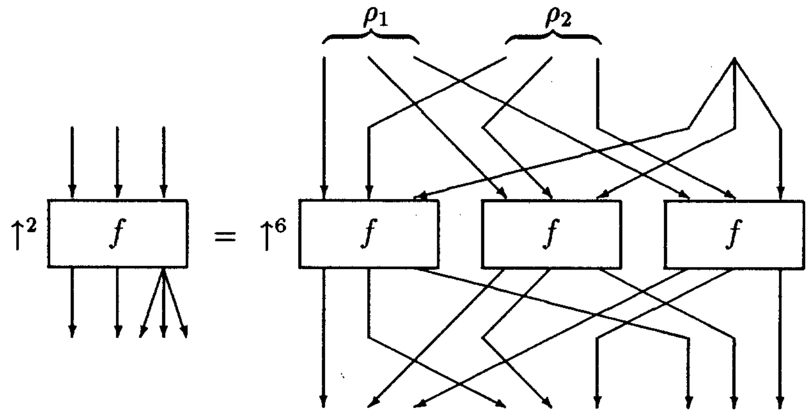

FIG. 1. The axiom C for $n=3$, $l=2$, and $p=q=1$.

The system IT above is closely related to the axioms used in (Ştefănescu, 1986; Căzănescu and Ştefănescu, 1988), and to the axiomatization appearing in (Ésik, 1982). It was proved in (Bartha, 1988) that $\mathscr{K}_S(\mathrm{IT})$ is equivalent to the variety of all iteration theories. (Recall that S denotes our basic algebra type.) It was also proved, cf. (Bartha, 1987a, 1987b), that $\mathscr{K}_D(\mathrm{DF} \cup \mathrm{TH})$ is equivalent to the variety of all algebraic theories. The S-algebras belonging to the variety $\mathscr{K}_S(\mathrm{FT})$ were given the name feedback theory after the obvious analogy.

Remark. The set of identities FT and IT are slightly redundant. For example, S5, S6, and S8, follow easily from TH. These axioms are present in FT and IT only because SF and SC are themselves relevant bases of identities, defining the varieties generated by all synchronous and ordinary scheme algebras, respectively. For more details, see (Bartha, 1987a, 1987b; Bloom and Ésik, 1985; Ştefănescu, 1986).

Note that SC is stronger than SF, because S7, S8, and S9 are consequences of SC. Therefore every iteration theory is a feedback theory and conversely, a feedback theory is an iteration theory iff it satisfies the identity X.

3. Strong Feedback and Strong Iteration Theories

Strong iteration theories can be derived axiomatically from iteration theories by replacing the unfortunately too complicated commutativity axioms by a somewhat stronger (but much simpler) implication. The origins of this implication can be found already in (Ésik, 1980, axioms X and X′). In (Ştefănescu, 1987) and (Ésik, 1988) the corresponding axiom was denoted I4 and C′, respectively. We introduce the same implication in a restricted and very simple form $\mathrm{C_s}$ below, and prove that $\mathrm{C_s}$ together with the identities $\mathrm{IT}\setminus\mathrm{C}$ provides an equivalent axiomatization of the class of strong iteration theories:

$\mathrm{C_s}$: if $f\cdot(\varepsilon_k+q)=(\varepsilon_k+p)\cdot g$ for some $k\in N$, $f\colon k+p\to k+q$, $g\colon 1+p\to 1+q$, then $\uparrow^k f=\uparrow g$.

$\mathrm{C_s}$ can be put in an even more comprehensive form as follows. Assume that for some $f\colon k+p\to k+q$ we have $\pi^i\cdot f\cdot(\varepsilon_k+q)=\pi^j\cdot f\cdot(\varepsilon_k+q)$ for all $i,j\in[k]$. Then

$$\uparrow^k f=\uparrow((\pi_k^i+p)\cdot f\cdot(\varepsilon_k+q)),$$

for any $i\in[k]$, of course. In other words, $\mathrm{C_s}$ expresses that the redundant computation $\uparrow^k f$ can be replaced by the more economical one $\uparrow g$.

Let $\mathrm{FT_s}$ and $\mathrm{IT_s}$ denote the set of axioms $\mathrm{FT}\setminus\mathrm{C}\cup\mathrm{C_s}$ and $\mathrm{IT}\setminus\mathrm{C}\cup\mathrm{C_s}$, respectively.

Definition 3.1. A strong feedback (strong iteration) theory is an S-algebra satisfying all the axioms $\mathrm{FT_s}$ (respectively, $\mathrm{IT_s}$).

Let $\mathrm{C'_s}$ be the following generalization of $\mathrm{C_s}$.

$\mathrm{C'_s}$: if $f\cdot(\beta+q)=(\beta+p)\cdot g$ for some $k,l\in N$, $f\colon k+p\to k+q$, $g\colon l+p\to l+q$ and mapping $\beta\colon k\to l$, then $\uparrow^k f=\uparrow^l g$.

Theorem 3.2. *$\mathrm{C'_s}$ is satisfied in all strong feedback and all strong iteration theories.*

Proof. First let $\beta=0_1$ in $\mathrm{C'_s}$ above. By assumption,

$$f\cdot(0_1+q)=0_1+f=(0_1+p)\cdot g.$$

Using TH we can write g into the form $\langle\mathrm{first}(g),\ \mathrm{tail}(g)\rangle$, where $\mathrm{first}(g)=\pi_{1+p}^1\cdot g$ and $\mathrm{tail}(g)=(0_1+p)\cdot g=0_1+f$. Consequently,

$$\begin{aligned}
\uparrow g &= \uparrow((\mathrm{first}(g)+0_1+f)\cdot w_2(1+q))\\
&\overset{\mathrm{(DF)}}{=} \uparrow((\mathrm{first}(g)+f)\cdot(1+w_2(q)))\\
&\overset{\mathrm{(S3)}}{=} \uparrow(\mathrm{first}(g)+f)\cdot w_2(q)\\
&\overset{\mathrm{(S1)}}{=} (\uparrow\mathrm{first}(g)+f)\cdot w_2(q)\\
&\overset{\mathrm{(TH)}}{=} f.
\end{aligned}$$

The reader is referred to (Ésik, 1982, Lemma 2) for an equivalent statement. Now let $\beta=\varepsilon_{i_1}+\cdots+\varepsilon_{i_l}$ for some nonnegative integers $i_1, ..., i_l$ such that $\sum_{j=1}^{l} i_j=k$. We prove $\uparrow^k f=\uparrow^l g$ by induction on l. If $l=0$, then $\beta=0$, so we have nothing to prove. The base step $l=1$ is already done. For $l\geqslant 2$ let $\beta'=\sum_{j=2}^{l}\varepsilon_{i_j}$, $q'=q+l-1$, and $p'=p+k-i_1$. Then we have

$$f\cdot(\beta+q)=(f\cdot(i_1+\beta'+q))\cdot(\varepsilon_{i_1}+q'),$$
$$(\beta+p)\cdot g=(\varepsilon_{i_1}+p')\cdot((1+\beta'+p)\cdot g).$$

Applying C_s we get that

$$\uparrow^{i_1}(f\cdot(i_1+\beta'+q))=\uparrow((1+\beta'+p)\cdot g).$$

Since the equations

$$\uparrow^{i_1}(f\cdot(i_1+\beta'+q))=(\uparrow^{i_1}f)\cdot(\beta'+q),$$
$$\uparrow((1+\beta'+p)\cdot g)=(\beta'+p)\cdot\uparrow g$$

are provable from SF, cf. (Bartha, 1987b, Claims S3* and S4*), we can apply the induction hypothesis for $\beta'\colon k-i_1\to l-1$ to obtain

$$\uparrow^k f=\uparrow^{k-i_1}(\uparrow^{i_1}f)=\uparrow^{l-1}(\uparrow g)=\uparrow^l g.$$

Finally, if $\beta\colon k\to l$ is arbitrary, then write β into the form $\alpha\cdot(\varepsilon_{i_1}+\cdots+\varepsilon_{i_l})$, where $\alpha\colon k\to k$ is a permutation and $\sum_{j=1}^{l} i_j=k$. By assumption,

$$(\alpha^{-1}+p)\cdot f\cdot(\alpha+q)\cdot\left(\sum\varepsilon_{i_j}+q\right)=\left(\sum\varepsilon_{i_j}+p\right)\cdot g.$$

Consequently, as we have seen above,

$$\uparrow^k((\alpha^{-1}+p)\cdot f\cdot(\alpha+q))=\uparrow^l g.$$

It remains to note that the equation $\uparrow^k f=\uparrow^k((\alpha^{-1}+p)\cdot f\cdot(\alpha+q))$ is provable from SF as shown in (Bartha, 1987b, Claim S2*).

Remark 3.3. Observe that if β is an injective mapping in C_s', then $\uparrow^k f=\uparrow^l g$ follows already from SF $\cup$ TH, since in this case the implication C_s is not used in the proof.

COROLLARY 3.4. *Every strong feedback* (*strong iteration*) *theory is a feedback* (*respectively, iteration*) *theory.*

Proof. It is easy to see that the identity C follows from C_s'. For a simple proof, see, e.g., (Ştefănescu, 1987).

COROLLARY 3.5. *The present definition of strong iteration theories is equivalent to the original one.*

Proof. According to (Bloom and Ésik, 1985; Ésik, 1988), a strong iteration theory is an algebraic theory satisfying the axioms SCH $\cup$ C′, where SCH is the set of scheme identities listed in (Bloom and Ésik, 1985) and C′ is the following implication:

C′: if $f: n \to n+q$, $g: m \to m+q$ and $\beta: n \to m$ are such that $f \cdot (\beta + q) = \beta \cdot g$, then $f^\dagger = \beta \cdot g^\dagger$.

Since SCH and SC are equivalent systems of identities, cf. (Bartha, 1987a), all we have to do is to turn $f^\dagger$ and $g^\dagger$ into the corresponding S-algebra expressions using the rewriting rule

$$f^\dagger = \uparrow^n (w_2(n) \cdot f) \qquad \text{for} \quad f: n \to n+q,$$

and check that C′ and C'_s are equivalent in any S-algebra satisfying SC $\cup$ TH. The obvious computations are left to the reader.

It is known, cf. (Ésik, 1988, Example), that strong iteration theories form a proper subclass of the variety of all iteration theories. This means at the same time that strong feedback theories, too, form a proper subclass of the variety of all feedback theories.

For a ranked alphabet Σ, let $\mathrm{Ft}(\Sigma)$ ($\mathrm{It}(\Sigma)$) denote the free feedback (respectively, free iteration) theory generated by Σ. It is known, cf. (Ştefănescu, 1987), that $\mathrm{It}(\Sigma)$ is a strong iteration theory. We shall represent $\mathrm{It}(\Sigma)$ in the usual way as an algebra of infinite regular $\Sigma_\perp$-trees, see (Elgot *et al.*, 1978; Wright *et al.*, 1976). To avoid ambiguity, we denote the feedback operation in $\mathrm{It}(\Sigma)$ by $\Uparrow$. In (Bartha, 1988) the following representation was given for $\mathrm{Ft}(\Sigma)$. Consider ∇ as a symbol of rank one in the alphabet $\Sigma_\nabla = \Sigma \cup \{\nabla\}$, and define $\mathrm{It}^\nabla(\Sigma)$ to be the S-algebra which is identical to the algebra

$$\mathrm{It}(\Sigma_\nabla)/\Omega(\Uparrow(\varepsilon \cdot \nabla) = \perp)$$

concerning its underlying sets and operations, except feedback. Feedback ($\uparrow$) is defined in $\mathrm{It}^\nabla(\Sigma)$ by the formula

$$\uparrow f = \Uparrow(f \cdot (\nabla + q)) \qquad \text{for} \quad f: 1+p \to 1+q.$$

Observe that $\boldsymbol{\alpha}(=\uparrow x) = \nabla$ and $\boldsymbol{\perp}(=\uparrow \varepsilon) = \perp$ in this algebra. It was proved in (Bartha, 1988) that $\mathrm{It}^\nabla(\Sigma)$ is a feedback theory, and it is isomorphic to $\mathrm{Ft}(\Sigma)$.

PROPOSITION 3.6. $\mathrm{Ft}(\Sigma)$ *is a strong feedback theory.*

Proof. Let $f: k+p \to k+q$, $g: 1+p \to 1+q$ in $\mathrm{Ft}(\Sigma)$ be such that $f \cdot (\varepsilon_k + q) = (\varepsilon_k + p) \cdot g$. By the above representation of $\mathrm{Ft}(\Sigma)$ we can assume that f and g are infinite trees in $\mathrm{It}(\Sigma_\nabla)$ not containing infinite ∇-branches. But then

$$f \cdot \left(\sum_{i=1}^{k} \nabla + q \right) \cdot (\varepsilon_k + q) = f \cdot (\varepsilon_k + q) \cdot (\nabla + q) = (\varepsilon_k + p) \cdot g \cdot (\nabla + q)$$

holds in the strong iteration theory $\mathrm{It}(\Sigma_\nabla)$, implying that $\uparrow^k f = \uparrow g$.

4. The Main Construction

Recall from (Gécseg and Peák, 1972) that a finite state structural Mealy automaton is defined as a six-tuple $A = (B, l, q, p, \delta, \lambda)$, where B is a finite set, l, p, q are nonnegative integers; $\delta: B^{l+q} \to B^l$ is the state transition function, and $\lambda: B^{l+q} \to B^p$ is the output function. Considering A as an ordinary automaton, B^l is the set of states, B^q and B^p are the input and output alphabet, respectively. The standard graphical representation of A is shown in Fig. 2, where the triangles symbolize the l state components (registers), and $f = \langle \delta, \lambda \rangle: B^{l+q} \to B^{l+p}$ is the combinational logic. We say that A is of sort $p \to q$. Instead of the six-tuple $(B, l, q, p, \delta, \lambda)$ we could use the more succinct notation $\uparrow^l f$ for A, which specifies both the number l of state components and the combinational logic $f: l+p \to l+q$ in the theory $T = \mathrm{Func}\, B$ of all functions over B. This idea is expressed in the definition of finite state structural T-automata (over an arbitrary theory T) given below.

The expression $A = \uparrow^l f$ is in close connection with Ştefănescu's (1986, 1987) and Căzănescu and Ştefănescu's (1988) Σ-flownomials (Σ-flowcharts

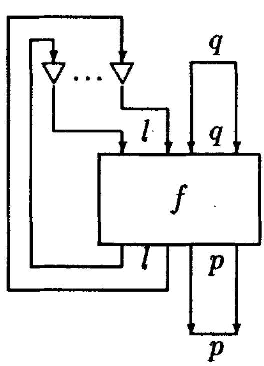

FIG. 2. An ST-automaton of sort $p \to q$.

in normal form) over T, where Σ is a doubly ranked alphabet and T is an appropriate theory with iteration or feedback. Such a flownomial of sort $p \to q$ is an expression of the form

$$F = \uparrow^n \left(\left(\sum_{i=1}^{k} \sigma_i + p \right) \cdot c \right),$$

where $\sigma_i \in \Sigma(n_i, m_i)$ $(i \in [k])$, $n = \sum_{i=1}^{k} n_i$, $m = \sum_{i=1}^{k} m_i$, and $c: m + p \to n + q$ is a morphism in T. Since F is a polynomial, it has a concrete value in $T(p, q)$ under every sort preserving valuation of the variables (boxes) σ_i. The morphism c specifies the *connectives* among the boxes in F. Note that the common origins of flownomials and scheme normal forms defined, e.g., in (Bloom and Ésik, 1985; Bartha, 1987a, 1987b) are Elgot's (1975) normal descriptions of flowchart schemes.

The relation between the structural automaton A and the flownomial F above is not obvious for the first sight, because the expression $\uparrow^l f$ does not contain any variables. This is due to the different meaning of $\uparrow$ in the two cases. In F, the application of $\uparrow^n$ means a simple redirection of control, while in A, the application of $\uparrow^l$ *creates* the triangles (boxes of sort $1 \to 1$) in Fig. 2. These boxes, however, need not (and in fact cannot) be valuated in T, because their meaning is *fixed* (though not expressible in the theory T). They represent the registers of the automaton A. To interpret A as a flownomial we should write it as $\uparrow^l((\sum_{i=1}^{l} \nabla + p) \cdot f)$, but this would no longer be a polynomial in our intended algebra because of the misuse of the feedback operation in it. Furthermore, we do not require T to be a theory with feedback, but define the exact meaning of the expression $\uparrow^l f$ by adjoining feedback to T as a new operation.

Let us point out another semantical difference between flownomials and structural automata. A flownomial of sort $p \to q$ is interpreted as a flowchart algorithm with p entries and q exits; see (Elgot, 1975; Ştefănescu, 1987). Accordingly, the flow of information in the flownomial (flowchart) follows the direction of the arrow between p and q. For example, the constant $\varepsilon: 2 \to 1$ should be interpreted as a join of two different paths in the flowchart. In a logical circuit, however, the meaning of ε is a branch; thus, in this case the information flows right in the opposite direction. Reasoning from the point of view of category theory the difference is the following. Concerning flownomials, the object n in the theory T is treated as the nth *copower* of the object 1, while in the case of structural T-automata n would rather be the nth *power* of 1, as in the original definition of Lawvere (1963). However, if we worked in this category, then the sort of a mapping $[p] \to [q]$ would confusingly become $q \to p$. Since mappings take a crucial part in our construction, we rather adhere to the coproduct formalism, and express the product-like (functional) semantics only by designing our schemes in an upside-down fashion; see, e.g., Fig. 3. One more reason to choose the coproduct formalism is to be in accordance with (Manes, 1976) in this section.

From the technical point of view, however, the present construction of the strong feedback theory $F_s T$ is similar to Ştefănescu's (1987) construction of the flowchart theory $\mathrm{Fl}_{\Sigma,T}/\equiv_d$ in the special case when the alphabet Σ contains only a single symbol ∇: $1 \to 1$. This is due to the syntactical connection between flownomials and structural automata described above, and to the fact that the algebraic properties of $\mathrm{Fl}_{\Sigma,T}/\equiv_d$ do not take part in the proof of Theorem 3.8 in (Ştefănescu, 1987). We emphasize this observation by separating the non-algebraic and algebraic features of the construction, discussing them in different sections, i.e., in this section and in the next one. Following Ştefănescu, we are going to operate with simple reduction steps (simulations via injective and surjective mappings) to obtain the desired minimal automaton. Simulations in the sense of Ştefănescu were first used in (Ésik, 1980).

Let us fix an algebraic theory T for Sections 4 and 5. As a first step we define the S-algebra Aut_T, which we call the algebra of *finite state structural T-automata* (ST-automata, for short), as follows:

$$\mathrm{Aut}_T(p, q) = \bigcup (\{l\} \times T(l+p, l+q) \mid l \in N).$$

Observe that the role of the number l in $(l, f) \in \mathrm{Aut}_T(p, q)$ is to specify the numbers p, q. Figure 2 shows how to imagine $A = (l, f): p \to q$ as a finite state structural Mealy automaton which has l registers, and its combinational logic is given by the morphism $f: l+p \to l+q$ in T. The attribute "finite state" in the definition concerns only the number of registers; the theory T can of course be arbitrary.

The operations and constants are interpreted in Aut_T in the following way;

Composition: if $F = (l, f): p \to q$ and $G = (m, g): q \to r$ are ST-automata, then

$$F \cdot G = (l+m, (x\#(l, m)+p) \cdot (m+f) \cdot (x\#(m, l)+q) \cdot (l+g));$$

see Fig. 3.

Sum: the sum of ST-automata $F_1 = (l_1, f_1): p_1 \to q_1$ and $F_2 = (l_2, f_2): p_2 \to q_2$ is the automaton

$$F_1 + F_2 = (l_1 + l_2, (l_1 + x\#(l_2, p_1) + p_2) \cdot (f_1 + f_2) \cdot (l_1 + x\#(q_1, l_2) + q_2));$$

see Fig. 4.

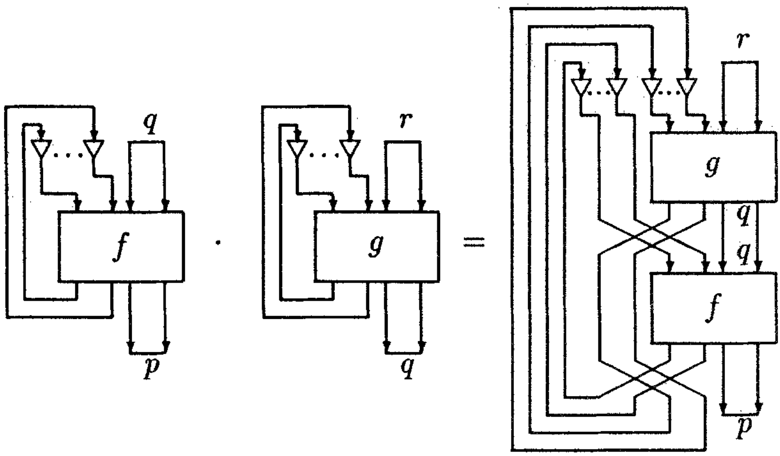

FIG. 3. Composition in Aut_T.

Feedback: if $F = (l, f)\colon 1 + p \to 1 + q$, then

$$\uparrow F = (l + 1, f);$$

see Fig. 5.

Constants: for each constant symbol c,

$$c_{\mathrm{Aut}_T} = (0, c_T).$$

For every $(p, q) \in N \times N$ we define a binary relation $\to$ (rather than $\to_{(p,q)}$) on the set $\mathrm{Aut}_T(p, q)$ in the following way.

DEFINITION 4.1. If $F = (l, f)$ and $G = (m, g)$ are ST-automata of sort $p \to q$, then $F \to G$ if there exists a mapping $\beta\colon l \to m$ such that

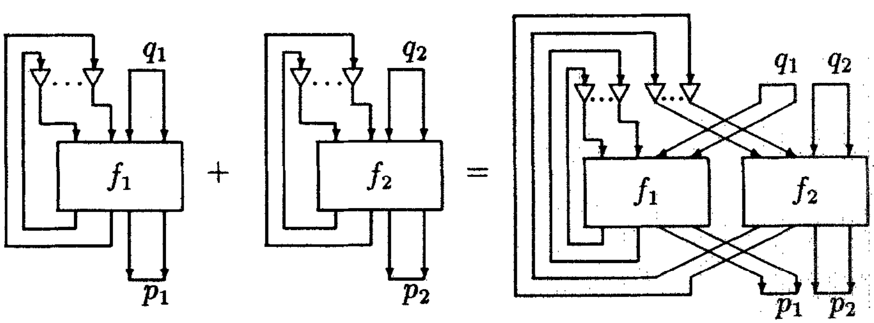

FIG. 4. Sum in Aut_T.

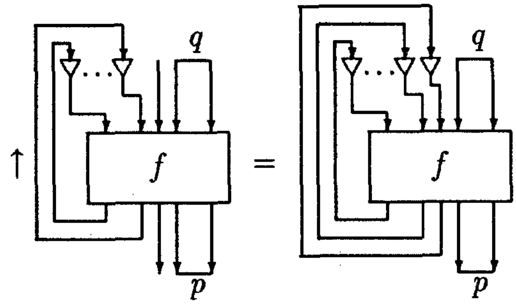

FIG. 5. Feedback in Aut_T.

$f \cdot (\beta + q) = (\beta + p) \cdot g$. In this case we say that F *simulates* G via the mapping β.

We adopt the notation $F \to_\beta G$ from (Ştefănescu, 1987) to specify the mapping β in hand. (Of course, β need not be unique in general.) We write $\to^s$ and $\to^i$ to mean that the simulation exists via a surjective (respectively, injective) mapping. The relation $\to^i \cap \to^s$ is denoted as $\leftrightarrow$ for obvious reasons. If $F \leftrightarrow G$, then we say that F and G are *isomorphic*.

Let $F = (l, f)$, $G = (m, g)$ and $H = (n, h)$ be ST-automata of sort $p \to q$ such that $F \to_\beta G \to_\gamma H$ for appropriate mappings β, γ. The commutative diagram of Fig. 6 shows that $\mathrm{Aut}_T(p, q)$ is itself a category for every $(p, q) \in N \times N$. The morphisms (also called 2-cells) of this category are mappings, which can be identified with the simulations induced by them. For $F = (l, f)$, $\mathrm{id}_F = l$ $(= \mathrm{id}_{[l]})$, and composition of 2-cells is defined as that of mappings.

$$\begin{tikzcd}
l+p \arrow[r, "f"] \arrow[d, "\beta+p"'] & l+q \arrow[d, "\beta+q"] \\
m+p \arrow[r, "g"] \arrow[d, "\gamma+p"'] & m+q \arrow[d, "\gamma+q"] \\
n+p \arrow[r, "h"] & n+q
\end{tikzcd}$$

FIG. 6. $\mathrm{Aut}_T(p, q)$ is a category.

The proofs of the following three simple observations are left to the reader.

PROPOSITION 4.2. *Let F, G, F', and G' be ST-automata of appropriate sorts such that $F \rightarrow_{\beta} F'$ and $G \rightarrow_{\gamma} G'$. Then*

(i) $F \cdot G \rightarrow_{\beta+\gamma} F' \cdot G'$;

(ii) $F + G \rightarrow_{\beta+\gamma} F' + G'$;

(iii) $\uparrow F \rightarrow_{\beta+1} \uparrow F'$.

Observation (i) implies that Aut_T is a 2-category (Kelly and Street, 1974) over the objects N. Indeed, if F, F', F'', G, G', G'' and β, β', γ, γ' are such that $F \rightarrow_{\beta} F' \rightarrow_{\beta'} F''$, $G \rightarrow_{\gamma} G' \rightarrow_{\gamma'} G''$, and F and G are composable, then

$$(\beta+\gamma) \cdot (\beta' + \gamma') = \beta \cdot \beta' + \gamma \cdot \gamma'.$$

Also, $\mathrm{id}_F + \mathrm{id}_G = \mathrm{id}_{F \cdot G}$, which is obvious. Note that in the strict sense $+$ is not a bifunctor in Aut_T, because we have only

$$(F_1 + F_2) \cdot (G_1 + G_2) \leftrightarrow F_1 \cdot G_1 + F_2 \cdot G_2$$

in it whenever the right side is defined. On the other hand it is clear that

$$(F + G) + H \leftrightarrow F + (G + H).$$

Thus, by (ii) of Proposition 4.2, Aut_T is a monoidal category up to isomorphism of 2-cells. It is also easy to verify that Aut_T is symmetric (again, only up to isomorphism of 2-cells).

Now we fit structural T-automata into the general framework of machines in a category by Arbib and Manes (1974a). Recall that a *process* in an arbitrary category $\mathscr{K}$ is an endofunctor $X: \mathscr{K} \rightarrow \mathscr{K}$. The category $\mathrm{Dyn}(X)$ of *X-dynamics* has as objects all pairs (Q, δ), where $\delta: XQ \rightarrow Q$ in $\mathscr{K}$, and as morphisms $(Q, \delta) \rightarrow (Q', \delta')$ all $\mathscr{K}$-morphisms $h: Q \rightarrow Q'$, rendering the diagram in Fig. 7 commutative. Identities and composition are

$$\begin{tikzcd}
XQ \arrow[r, "\delta"] \arrow[d, "X\gamma"'] & Q \arrow[d, "\gamma"] \\
XQ' \arrow[r, "\delta'"] & Q'
\end{tikzcd}$$

FIG. 7. A morphism $(Q, \delta) \rightarrow (Q', \delta')$ in $\mathrm{Dyn}(X)$.

defined in $\mathrm{Dyn}(X)$ in the natural "algebraic" way; see (Arbib and Manes, 1974a). Often we identify the dynamics (Q, δ) with the morphism $\delta: XQ \to Q$ if Q is understood. The morphisms of $\mathrm{Dyn}(X)$ are called *dynamorphisms.* An *X-automaton* is a 6-tuple $(Q, \delta, I, \tau, Y, \beta)$, where (Q, δ) is an X-dynamics and $\tau: I \to Q$, $\beta: Q \to Y$ are mophisms in $\mathscr{K}$. This definition generalizes the classical Moore automaton concept, which can be recaptured by setting $\mathscr{K} = \mathbf{Set}$ (i.e., the category of all sets and mappings), using $\_ \times X: \mathbf{Set} \to \mathbf{Set}$ for the process and setting $I = 1$. In general, Q, I, and Y are called the state object, the initial object, and the output object, respectively; τ is the initial state and β is the output morphism.

In our case $\mathscr{K} = T^{\mathrm{op}}$ (i.e., the opposite of T) and X is the functor $\mathbf{q} = \_ + q$ for some fixed $q \in N$. Observe that, in terms of the theory T, $\mathbf{q}$-dynamics are rather codynamics, i.e., pairs (l, f) with $f: l \to l + q$. To treat ST-automata of sort $p \to q$ as appropriate $\mathbf{q}$-dynamics, we consider state objects of the form $l + p$ only, for a fixed $p \in N$. Then a p-allowable $\mathbf{q}$-dynamics would be of sort $l + p \to l + p + q$, but in this way we have produced a superfluous p on the right-hand side. Again, we restrict the scope of p-allowable dynamics by imposing that the morphism $f: l + p \to l + p + q$ in T must not *depend on* (the variables) $\{l + 1, ..., l + p\}$. To explain this last concept we have to make a short digression.

Since in this section we rely mainly on (Manes, 1976), it is necessary to synchronize between algebraic theories in the sense of Lawvere and those in the more general sense of Manes. This was done in full details in Section 1.5 of (Manes, 1976), here we just highlight the most important points. An algebraic theory in the sense of Manes is in fact a *triple* (or monad); see also (MacLane, 1971). Our theory T (in the sense of Lawvere) can be derived from a suitable *finitary* algebraic theory $\mathbf{T}$ (in the triple sense) in $\mathbf{Set}$ by restricting the *Kleisli category* of $\mathbf{T}$ to the objects $[n]$, $n \in N$. This category is called the *Lawvere theory* of $\mathbf{T}$ in (Manes, 1976).

A morphism $f: 1 \to q$ in T (i.e., $f \in \mathbf{T}[q]$) is called a syntactic operation. The *arity* of f is the smallest number n such that $f = f^{\#} \cdot \alpha$ for some $f^{\#}: 1 \to n$ in T and *injective* mapping $\alpha: n \to q$. Let $\mathrm{sp}(f) \subseteq [q]$ denote the image of α. (Note the ambiguity of this notation, which will be explained immediately.) According to the semantic interpretation of f described in (Manes, 1976), the following two cases are possible.

1. $n = 1$ and $\mathrm{sp}(f \cdot (q + 0_1))$ is not uniquely determined. (Note that q may be 1 as well.) In this case f is a constant, but there are no true constants in T, i.e., $T(1, 0) = \varnothing$.

2. $\mathrm{sp}(f) = \mathrm{sp}(f \cdot (q + 0_1))$ is uniquely determined. In this case we say that f depends on each element of $\mathrm{sp}(f)$ (and only on these).

In both cases, however, $f^{\#}$ is unique up to isomorphism. This morphism is called the *core* of f, while $\mathrm{sp}(f)$ is the *minimal support* of f if f is not a constant. The minimal support of a constant is evidently empty. If case 1 never occurs, then we say that T is (algebraically) *closed.* Thus, T is closed iff every constant in T is a true constant. Note that among the two trivial algebraic theories of (Manes, 1976) only the larger one is closed. The initial theory is an example of a closed theory satisfying $T(1, 0) = \varnothing$.

Every theory T has an obvious closure $\bar{T}$ for which

$$\bar{T}(1, 0) = \{f \in T(1, 1) \mid f \text{ is a constant}\}$$

if $T(1, 0) = \varnothing$.

For $f: p \to q$ in T, we define $\mathrm{sp}(f) = \bigcup_{i=1}^{p} \mathrm{sp}(\pi_p^i \cdot f)$, and say that f depends on $j \in [q]$ if so does $\pi_p^i \cdot f$ for some $i \in [p]$. The following statement is obvious.

LEMMA 4.3. *Let f: $p \to q$ be a morphism in T and γ: $p' \to p$, α: $q \to q'$ be mappings. If γ is onto, then* $\mathrm{sp}(\gamma \cdot f \cdot \alpha) \subseteq \alpha(\mathrm{sp}(f))$. *If $p = 1$ and α is injective, then f and $f \cdot \alpha$ have the same arity with the same core.*

LEMMA 4.4. *Let α_1: $p_1 \to q$ and α_2: $p_2 \to q$ be a pair of injective mappings such that their images are not disjoint. Then the pullback of α_1 and α_2 in* **Set** *is a pullback in T as well.*

Proof. Let $\beta_1: n \to p_1$, $\beta_2: n \to p_2$ be the pullback of (α_1, α_2) in **Set** (note that $n \geqslant 1$). We have to prove that if $f_1: m \to p_1$ and $f_2: m \to p_2$ are such that $f_1 \cdot \alpha_1 = f_2 \cdot \alpha_2$, then there exists a unique $f: m \to n$ for which $f_1 = f \cdot \beta_1$ and $f_2 = f \cdot \beta_2$. Without loss of generality we can assume that $m = 1$. By Lemma 4.3, f_1 and f_2 have the same arity k. Let $f_1 = f^{\#} \cdot \gamma_1$ and $f_2 = f^{\#} \cdot \gamma_2$ for appropriate injections γ_1, γ_2, where $f^{\#}$ is the core of f_1 and f_2 (i.e., the common core of $f_1 \cdot \alpha_1$ and $f_2 \cdot \alpha_2$, which is the same). If T is closed or f is not a constant, then $\gamma_1 \cdot \alpha_1 = \gamma_2 \cdot \alpha_2$ follows immediately, even without the assumption that the images of α_1 and α_2 are disjoint. Otherwise it may be that f is a constant, $k = 1$, and γ_1, γ_2 are arbitary. In this case set $\gamma_1(1) = i$ and $\gamma_2(1) = j$ so that $\alpha_1(i) = \alpha_2(j)$, to obtain $\gamma_1 \cdot \alpha_1 = \gamma_2 \cdot \alpha_2$ again. Since (β_1, β_2) is pullback, there exists a unique $\gamma: k \to n$ such that $\gamma_1 = \gamma \cdot \beta_1$ and $\gamma_2 = \gamma \cdot \beta_2$. The rest of the proof can be read out from the commutative diagram of Fig. 8.

Returning to the interpretation of structural T-automata as machines in the category T^{op}, let us make clear the role of the output morphism β: $Q \to Y$ and the initial state τ: $I \to Q$. It is evident that Y should be the object p and $\beta = 0_l + p$: $p \to l + p$ in T. Regarding τ, our philosophy is the following. Since data objects are supposed to be morphisms of sort $1 \to 0$, we fix $I = 0$. It might be the case, however, that $T(1, 0) = \varnothing$, besides we do

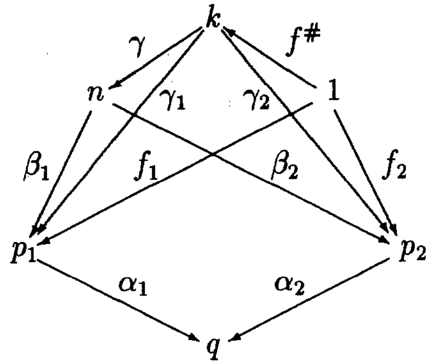

FIG. 8. The proof of Lemma 4.4 in a diagram.

not want to distinguish any particular morphism $l+p \to 0$ in T as being initial. Instead, we take the free extension of T to a pointed theory $T_\perp$ and define τ to be $\sum_{i=1}^{l+p} \perp$ in that theory. (Recall from (Bartha, 1989) that a pointed theory is a theory equipped with a further constant $\perp: 1 \to 0$.)

Under these assumptions a **q**-automaton $M=(l+p, f, 0, \tau, p, \beta)$ with a p-allowable (co-) dynamics $f: l+p \to l+p+q$ runs in the following expected way. Its real initial state is $(l+0_p)\cdot\tau: l \to 0$. For each current state $s: l \to 0$ and current input $r: q \to 0$ in $T_\perp$, it first computes

$$\text{step}(s, r) = f\cdot\left(s + \sum_{i=1}^{p} \perp + r\right): l+p \to 0,$$

then sets its new state to $(l+0_p)\cdot\text{step}(s, r)$ and emits $\beta\cdot\text{step}(s, r)$ as current output. The machine runs forever, producing an infinite sequence of output data vectors responding to an infinite sequence of input data vectors. Note that this is a major difference between the interpretation of structural T-automata and that of linear systems by (Padulo and Arbib, 1974; Arbib and Manes, 1974b). In that category theoretical model of linear systems the input sequence is cut down to finite length in order to provide a left adjoint for the functor $U: \text{Dyn}(X) \to \mathscr{K}$ which forgets the dynamics associated with a given state object, i.e., to make X an input process. For a state object Q, the state object $X^{@}Q$ of the free dynamics over Q is then obtained as the infinite coproduct of Q, XQ, X^2Q, ..., analogously to the classical construction. In our case this analogy does not work, because we do not have products in T (note the duality). Extending the scope of the theory T from the Kleisli category to the whole Eilenberg–Moore category of the corresponding triple **T** would be a solution to this problem, but then how do we define the process **q**? In the present setting **q** fails to be input or output in the sense of (Arbib and Manes, 1974a), although it has straightforward input–output behavior.

$$\begin{tikzcd} & Q \arrow[dd, "\psi"] \arrow[dr, "\beta"] & \\ I \arrow[ur, "\tau"] \arrow[dr, "\tau'"'] & & Y \\ & Q' \arrow[ur, "\beta'"'] & \end{tikzcd}$$

FIG. 9. Simulation between X-automata.

Let $M=(Q, \delta, I, \tau, Y, \beta)$ and $M'=(Q', \delta', I, \tau', Y, \beta')$ be two X-automata. Recall from (Arbib and Manes, 1974a) that a simulation from M to M' is a dynamorphism $\psi: (Q, \delta) \to (Q', \delta')$ which commutes with input and output as shown in Fig. 9.

In our case $Q=l+p$, $Q'=m+p$, $\beta=0_l+p$, $\beta'=0_m+p$ for some $l, m \in N$. It follows immediately from the right side of the diagram in Fig. 9 that $\psi=h+p$ for some $h: m \to l$. However, the left side of the diagram does not require that h be a mapping. This is only an obvious sufficient condition to make the diagram commutative Nevertheless, in this paper we consider mappings only as simulations, and call them *simple* simulations.

Remark 4.5. Since the general theory of (Arbib and Manes, 1974a) does not apply now, we do not know whether simulations are indeed respected by the i/o behavior of the automata, i.e., whether M and M' have the same behavior or not. A positive answer to this question can be obtained easily in an intuitive way, but for a more substantial proof see (Bartha, 1990). In that paper the i/o behavior of structural T-automata was modeled in a further extension of the pointed theory $T_\perp$ to a strong feedback theory $F_1^\infty T_\perp$, where F_1^∞ is an appropriate functor; see also (Bartha, 1989). In this paper we characterize the strong feedback theory $F_s T$ freely generated by T by a minimization process of the same automata. It would be nice to have $F_1^\infty T_\perp \cong F_s T$, but this cannot be expected in general because in the present minimization we are dealing with simple simulations (i.e., mappings) only.

Let $\mathrm{Aut}_T^{\ddagger}(p, q)$ denote the category which has as objects all $\mathbf{q}$-automata with p-allowable $\mathbf{q}$-dynamics and as morphisms simple simulations between the automata. If $f: l+p \to l+p+q$ is p-allowable, then there exists a unique morphism $\bar{f}: l+p \to l+q$ in $\bar{T}$ such that (in $\bar{T}$) $f=\bar{f} \cdot (l+0_p+q)$. (Recall that $\bar{T}$ is the closure of T.) Similarly, for every simple simulation $(l+p, f) \to^{\alpha} (m+p, g)$ in $\mathrm{Aut}_T^{\ddagger}(p, q)$ there exists a unique arrow $(l, \bar{f}) \to_{\bar{\alpha}} (m, \bar{g})$ in $\mathrm{Aut}_T(p, q)$ such that $\alpha=\bar{\alpha}+p$. The following statement should now be obvious.

PROPOSITION 4.6. *The categories* $\mathrm{Aut}_T^*(p, q)$ *and* $\mathrm{Aut}_T(p, q)$ *are isomorphic for every* $(p, q) \in N \times N$. *The isomorphism is determined by the connection* $f \mapsto \bar{f}$, $\alpha \mapsto \bar{\alpha}$ *above.*

Let $U^*(p, q)$: $\mathrm{Aut}_T^*(p, q) \to T$ denote the restriction of the natural underlying T-object functor U: $\mathrm{Dyn}(\mathbf{q}) \to T$: $(l, f) \mapsto l$ to p-allowable dynamics and simple simulations.

PROPOSITION 4.7. $U^*(p, q)$ *creates pushouts.*

Proof. It is known that U creates colimits for all diagrams in $\mathrm{Dyn}(\mathbf{q})$. For a dual statement, see, e.g., (Manes, 1976, Proposition 3.1.19). It is also clear that the pushout of two mappings in T is the same as their pushout in **Set**. It remains to prove that the created pushout diagram is in $\mathrm{Aut}_T^*(p, q)$. To this end suppose that

$$G_1 \leftarrow^{\beta_1 + p} F \rightarrow^{\beta_2 + p} G_2$$

for some $F = (l + p, f)$, $G_i = (m_i + p, g_i)$, $i = 1, 2$ in $\mathrm{Aut}_T^*(p, q)$. With $\rho_1 = m_1 + p + 0_{m_2 + p}$ and $\rho_2 = 0_{m_1 + p} + m_2 + p$ construct the coproduct dynamics

$$G' = (m_1 + p + m_2 + p, \langle g_1 \cdot (\rho_1 + q), g_2 \cdot (\rho_2 + q) \rangle)$$

of G_1 and G_2 (which is clearly not p-allowable) and consider the pair of parallel dynamorphisms $\gamma_i' = (\beta_i + p) \cdot \rho_i$ between F and G'. Let $G' \rightarrow^{\gamma} G$ be the coequalizer of γ_1' and γ_2'. It is known, cf. (MacLane, 1971), that γ is surjective and the desired pushout diagram is

$$G_1 \rightarrow^{\gamma_1} G \leftarrow^{\gamma_2} G_2,$$

where $\gamma_i = \rho_i \cdot \gamma_i$. Since G' does not depend on $\{m_i + 1, ..., m_i + p\}$ for either $i = 1$ or $i = 2$, Lemma 4.3 implies that G is p-allowable. Moreover, γ_1 and γ_2 are both simple simulations, which is obvious.

PROPOSITION 4.8. $U^*(p, q)$ *creates pullbacks for all pairs of injective simple simulations.*

Proof. Let

$$G_1 \rightarrow^{\alpha_1 + p} F \leftarrow^{\alpha_2 + p} G_2$$

for $F = (m + p, f)$, $G_i = (l_i + p, g_i)$, $i = 1, 2$ and injective mappings α_i: $l_i \to m$. If the images of α_1 and α_2 are disjoint, then clearly $p = 0$ and the desired pullback is $G_1 \leftarrow^{0_{l_1}} (0, 0_q) \rightarrow^{0_{l_2}} G_2$. If this is not the case, then we "switch" categories by considering $\mathbf{q}$-dynamics to be objects in the arrow-only category T^2 (MacLane, 1971). In this setting a dynamorphism becomes the first component of a morphism in T^2. It is known, cf. (MacLane, 1971, Exercise V.1.3), that the projection functor $P\colon T^2 \to T \times T$ which sends each morphism $n \to m$ in T to the pair (n, m) creates limits. (Note that P creates colimits as well, so we could have applied this trick already in the proof of Proposition 4.7, knowing that $\mathbf{q}$ preserves colimits.) Let (β_1, β_2) be the pullback of (α_1, α_2), which exists by Lemma 4.4. Clearly, the pullbacks of $(\alpha_1 + p,\ \alpha_2 + p)$ and $(\alpha_1 + p + q,\ \alpha_2 + p + q)$ are $(\beta_1 + p,\ \beta_2 + p)$ and $(\beta_1 + p + q,\ \beta_2 + p + q)$, respectively, so that P is able to create a pullback

$$G_1 \leftarrow^{\beta_1 + p} G \rightarrow^{\beta_2 + p} G_2.$$

By Lemma 4.3 G is p-allowable.

Corollary 4.9. *For every $(p, q) \in N \times N$, $\mathrm{Aut}_{\bar{T}}(p, q)$ has all pushouts. It also has pullbacks of all pairs of simulations via injective mappings. The pushout and pullback arrows are obtained in the same way as in* **Set**.

The following example shows that Corollary 4.9 is no longer true in $\mathrm{Aut}_T(p, q)$.

Example. Let $f \in T(1, 1)$ be a constant which is not a true constant. Then for $F_1 = (1, \varepsilon \cdot f)$ and $F_2 = (1, (1 + f) \cdot \varepsilon)$ we have

$$F_1 \rightarrow_{\pi_2^2} (2, 1 + \varepsilon \cdot f) = (2, (1 + \varepsilon \cdot f) \cdot \varepsilon + 0_1) \leftarrow_{\pi_2^1} F_2.$$

It is, however, impossible to create a pullback in this situation, because $T(1, 0) = \varnothing$ and the pullback of (π_2^2, π_2^1) in **Set** is $(0_1, 0_1)$.

From now on we abandon the categories $\mathrm{Aut}_T^*(p, q)$, and continue working in the (isomorphic) categories $\mathrm{Aut}_{\bar{T}}(p, q)$. Let us agree that, in the sequel, by an ST-automaton $p \to q$ we mean an object in $\mathrm{Aut}_{\bar{T}}(p, q)$ rather than an object in $\mathrm{Aut}_T(p, q)$ as we have meant so far.

Proposition 4.10. *In $\mathrm{Aut}_{\bar{T}}(p, q)$ every arrow has an epi–mono factorization.*

Proof. Supposing that $(l, f) = F \rightarrow_\beta G = (m, g)$, let $\beta = \alpha \cdot \gamma$ be the surjective–injective factorization of the mapping β with $\alpha\colon l \to n$ and $\gamma\colon n \to m$. Let α_{-1} (γ^{-1}) be any right (left) inverse of α (respectively, γ), and put

$$H = (n, (\gamma + p) \cdot g \cdot (\gamma^{-1} + q)).$$

$$
\begin{tikzcd}
l+p \arrow[r, "f"] \arrow[d, shift right, "\alpha+p"'] & l+q \arrow[d, "\alpha+q"] \\
n+p \arrow[u, dashed, shift right, "\alpha_{-1}+p"'] \arrow[r, dashed] \arrow[d, "\gamma+p"'] & n+q \arrow[d, shift left, "\gamma+q"] \\
m+p \arrow[r, "g"'] & m+q \arrow[u, dashed, shift left, "\gamma^{-1}+q"]
\end{tikzcd}
$$

FIG. 10. Epi–mono factorization of an arrow in $\mathrm{Aut}_T(p, q)$.

It is immediate from the diagram of Fig. 10 that $F \to_\alpha H$ and $H \to_\gamma G$. Since every simulation via a surjective (injective) mapping is a coequalizer (respectively, equalizer) in $\mathrm{Aut}_T(p, q)$, α is epi and γ is mono. It is known, cf. (Manes, 1976), that in this case H is unique up to isomorphism. In terms of (Manes, 1976), simulations via injective and surjective mappings form an image factorization system in $\mathrm{Aut}_T(p, q)$.

DEFINITIONS. An ST-automaton $F\colon p \to q$ is *accessible* if $G \to_\beta F$ for an injective mapping β implies that β is an isomorphism. F is *reduced* if $F \to_\beta G$ for surjective β implies that β is an isomorphism. If $F \to_\beta G$ for some ST-automata $F, G\colon p \to q$, then F is a *subautomaton* of G if β is injective. F is a *minimal* subautomaton of G if it is accessible as well. G is a *quotient of* F if β is surjective. In this case G is a *maximal* quotient if it is also reduced.

Explaining the terminology, it can be proved that an ST-automaton $F = (l, f)\colon p \to q$ is accessible iff for every $k \in [l]$ there exists a sequence $i_0, i_1, ..., i_m$ of nonnegative integers such that

(i) $l < i_0 \leqslant l + p$, $i_j \in [l]$ if $j \in [m]$ and $i_m = k$;
(ii) $\pi^{i_{j-1}} \cdot f$ depends on i_j for every $j \in [m]$.

A similar fact is known to hold for Σ-trees, cf. (Ésik, 1980).

By Corollary 4.9, every ST-automaton has a unique minimal subautomaton and a unique maximal quotient. If F is reduced, then every subautomaton of F is reduced. Indeed, if $F \leftarrow_\beta G \to_\gamma H$ with β and γ being injective and surjective, respectively, then in the corresponding pushout

$F \rightarrow_{\bar{\gamma}} \bar{G} \leftarrow_{\bar{\beta}} H$ $\bar{\beta}$ and $\bar{\gamma}$ are injective and surjective again; see (MacLane, 1971). Since F is reduced, $\bar{\gamma}$ (and so γ) is an isomorphism. Note, however, that the quotient of an accessible automaton need not remain accessible, because we do not have the corresponding pullback in $\mathrm{Aut}_T(p, q)$.

Let $\mathscr{K}$ be a category. Recall from (MacLane, 1971) that two objects $j, k \in \mathscr{K}$ are in the same *connected component* of $\mathscr{K}$ if there is a finite sequence of arrows

$$j = j_0 \rightarrow j_1 \leftarrow j_2 \rightarrow \cdots \rightarrow j_{2n-1} \leftarrow j_{2n} = k$$

joining j to k.

DEFINITION. Two ST-automata F, $G: p \rightarrow q$ are *equivalent*, in notation $F \sim G$, if they are contained in the same connected component of $\mathrm{Aut}_T(p, q)$.

THEOREM 4.11. *If $F \sim G$, then there exists an ST-automaton H such that $F \rightarrow H \leftarrow G$.*

Proof. By Corollary 4.9, the arrow $\rightarrow$ satisfies the so called diamond property. It follows then in the standard way, cf. (Barendregt, 1984, Theorem 3.1.12.), that $\sim \, = \, \rightarrow \circ \leftarrow$.

Now we are ready to prove the main result of this section, which parallels the Minimal Realization Theorem of (Arbib and Manes, 1974a).

THEOREM 4.12. *In every connected component $\mathscr{C}$ of* $\mathrm{Aut}_T(p, q)$ *there exists an automaton $M_{\mathscr{C}}$ which is both accessible and reduced, i.e., minimal regarding the number of its registers. Starting from any $F \in \mathscr{C}$, $M_{\mathscr{C}}$ can be obtained as the minimal subautomaton of the maximal quotient of F, therefore it is unique up to isomorphism.*

Proof. Since the property of being reduced is preserved in subautomata, it is sufficient to prove the second statement only. Let F_1 and F_2 be equivalent ST-automata in $\mathscr{C}$, and construct the corresponding minimal automata M_1 and M_2 as it is described in the theorem. The situation is shown by Fig. 11. Clearly $M_1 \sim M_2$, hence by Theorem 4.11 there exist H, β_1, and β_2 such that

$$M_1 \rightarrow_{\beta_1} H \leftarrow_{\beta_2} M_2.$$

Since M_i $(i = 1, 2)$ is reduced, the epi–mono factorization of the arrow $M_i \rightarrow_{\beta_i} H$ shows that β_i is injective. Thus, we have a pullback

$$M_1 \leftarrow_{\bar{\beta}_2} \bar{H} \rightarrow_{\bar{\beta}_1} M_2$$

$$\begin{tikzcd}
& & \bar{H} \arrow[dl, "\bar{\beta}_2"'] \arrow[dr, "\bar{\beta}_1"] & & \\
& M_1 \arrow[dr, "\beta_1"'] \arrow[dd, "\gamma_1"'] & & M_2 \arrow[dl, "\beta_2"] \arrow[dd, "\gamma_2"] & \\
& & H & & \\
F_1 \arrow[r, "\alpha_1"'] & G_1 & & G_2 & F_2 \arrow[l, "\alpha_2"]
\end{tikzcd}$$

FIG. 11. The proof of Theorem 4.12 in a diagram.

with $\bar{\beta}_1$ and $\bar{\beta}_2$ being injective, too. But then M_1 and M_2 are isomorphic, because they are both accessible.

5. THE STRONG FEEDBACK THEORY $F_s T$.

For the rest of the paper we return to the $N \times N$-sorted algebra interpretation of the categories T and $\mathrm{Aut}_{\bar{T}}$, and consider the arrow $\rightarrow$ in $\mathrm{Aut}_{\bar{T}}$ to be a relation, as it was introduced under 4.1. By Proposition 4.2, $\sim$ is a congruence relation of $\mathrm{Aut}_{\bar{T}}$. Let $F_s T$ denote the S-algebra $\mathrm{Aut}_{\bar{T}}/\sim$.

THEOREM 5.1. *$F_s T$ is a strong feedback theory.*

Proof. We reached to a point where we can benefit much from the simplicity of our axioms. In fact, none of the proofs showing them (except C_s) to be valid in $F_s T$ is more than a simple exercise. To provide two examples we prove S9 and TH here, leaving the rest to the reader.

Proof of S9. We prove only the case $n = 1$ now, because the rest follows from C_s. By definition, $\boldsymbol{\varepsilon} = (0, \varepsilon)$ and $\boldsymbol{\alpha} = \uparrow x = (1, x)$ in $\mathrm{Aut}_{\bar{T}}$. A straightforward computation shows that

$$\boldsymbol{\varepsilon} \cdot \boldsymbol{\alpha} = (1, (1 + \varepsilon) \cdot x).$$

Thus,

$$\uparrow(\boldsymbol{\varepsilon} \cdot \boldsymbol{\alpha}) = (2, (1 + \varepsilon) \cdot x) \rightarrow_{\varepsilon} (1, \varepsilon) = \uparrow \boldsymbol{\varepsilon} = \boldsymbol{\perp}.$$

Proof of TH. The case $p = 0$ postulates that $f = 0_q$ for all $f\colon 0 \rightarrow q$ (recall that $w_0(q) = 0_q$ and $0 \cdot f = f$). Indeed, if $G = (m, g)$ is an ST-automaton of sort $0 \rightarrow q$ (i.e., $g\colon m \rightarrow m + q$ in $\bar{T}$), then $(0, 0_q) \rightarrow_{0_m} (m, g)$. If $p \geqslant 1$, then let $F = (l, f)\colon p \rightarrow q$ be arbitrary. It is routine to check that

$$w_p(p) \cdot F = (l, (l + w_p(p)) \cdot f),$$

$$\left(\sum_{i=1}^{p} F\right) \cdot w_p(q) = (l \cdot p, f' \cdot (l \cdot p + w_p(q))),$$

where

$$f' = \kappa(p, 2) \# ((l)^p, (p)^p) \cdot \left(\sum_{i=1}^{p} f \right) \cdot \kappa(2, p) \# (l, q)^p.$$

A short computation in the theory $\bar{T}$ yields

$$\begin{aligned} f' \cdot (l \cdot p + w_p(q)) \cdot (w_p(l) + q) &= \kappa(p, 2) \# ((l)^p, (p)^p) \cdot \left(\sum_{i=1}^{p} f \right) \cdot w_p(l+q) \\ &= (w_p(l) + w_p(p)) \cdot f, \end{aligned}$$

showing that

$$(l \cdot p, f' \cdot (l \cdot p + w_p(q))) \rightarrow_{w_p(l)} (l, (l + w_p(p)) \cdot f).$$

Concerning C_s we have to prove that if $F = (l, f)\colon k + p \rightarrow k + q$ and $G = (m, g)\colon 1 + p \rightarrow 1 + q$ are such that $F \cdot (\varepsilon_k + q) \sim (\varepsilon_k + p) \cdot G$, then $\uparrow^k F \sim \uparrow G$. As above,

$$F \cdot (\varepsilon_k + q) = (l, f \cdot (l + \varepsilon_k + q)),$$

$$(\varepsilon_k + p) \cdot G = (m, (m + \varepsilon_k + p) \cdot g),$$

hence by Theorem 4.11 there exist $H = (n, h)\colon k + p \rightarrow 1 + q$ and mappings $\beta\colon l \rightarrow n$, $\gamma\colon m \rightarrow n$ for which

$$f \cdot (l + \varepsilon_k + q) \cdot (\beta + 1 + q) = (\beta + k + p) \cdot h; \tag{1}$$

$$(m + \varepsilon_k + p) \cdot g \cdot (\gamma + 1 + q) = (\gamma + k + p) \cdot h. \tag{2}$$

By (2), $\pi^i \cdot h = \pi^j \cdot h$ for all $n < i, j \leqslant n + k$, so that h can be written in the form $(n + \varepsilon_k + p) \cdot h'$ for an appropriate morphism $h'\colon n + 1 + p \rightarrow n + 1 + q$ in T. Consider the ST-automaton $H' = (n, h')\colon 1 + p \rightarrow 1 + q$. Using (1),

$$f \cdot (\beta + \varepsilon_k + q) = (\beta + \varepsilon_k + p) \cdot h',$$

that is, $\uparrow^k F \rightarrow_{\beta + \varepsilon_k} \uparrow H'$. On the other hand,

$$(\varepsilon_k + p) \cdot G \sim H = (\varepsilon_k + p) \cdot H',$$

implying that $G \sim H'$ and so $\uparrow G \sim \uparrow H'$. Thus, $\uparrow^k F \sim \uparrow G$, which was to be proved.

THEOREM 5.2. *$F_s T$ is the strong feedback theory freely generated by T.*

Proof. Both theories T and $\bar{T}$ can be embedded into $\mathrm{Aut}_{\bar{T}}$ by the mapping $\iota\colon f \mapsto (0, f)$, which is clearly injective. It is also clear that the quotient of ι by $\sim$ is still an embedding of $\bar{T}$ into $F_s T$. $F_s T$ is generated by $\bar{T}$ because if $F=(l, f)\colon p \to q$ is an arbitrary ST-automaton, then

$$F = \uparrow^l(\iota(f)) \tag{3}$$

holds by definition. Moreover, $\bar{T}$ is generated by T in $F_s T$. Indeed, if T is not closed and $f\colon 1 \to 1$ is a constant in T, then

$$(0, f^{\#}) \to_{0_1} (1, \varepsilon \cdot f) = \uparrow(0, \varepsilon \cdot f),$$

where $f^{\#}$ is the true constant in $\bar{T}$ corresponding to f. It remains to show that, given a strong feedback theory A and a theory map (D-algebra homomorphism) $\phi\colon T \to A$, ϕ can be extended in a unique way to a homomorphism $\bar{\phi}\colon F_s T \to A$. Let $F=(l, f)$ be an ST-automaton $p \to q$, and let $F\sim$ denote the equivalence class of $\sim$ (connected component of $\mathrm{Aut}_T(p, q)$) containing F. By (3) we are forced to define

$$\bar{\phi}(F\sim) = \uparrow^l(\phi(f)),$$

so $\bar{\phi}$ becomes a unique extension of ϕ. If $F \sim G$, then by Theorem 4.11 there exist mappings β, γ and an ST-automaton H such that $F \to_\beta H$ and $G \to_\beta H$. Since A is a strong feedback theory, Theorem 3.2 implies that $\bar{\phi}$ is well-defined. To show that $\bar{\phi}$ is a homomorphism we prove $\bar{\phi}(F_1 + F_2) = \bar{\phi}(F_1) + \bar{\phi}(F_2)$ only, because the same reasoning applies for the other two operations as well, and the constants are clearly preserved by $\bar{\phi}$. If $F_1 = (l_1, f_1)\colon p_1 \to q_1$ and $F_2 = (l_2, f_2)\colon p_2 \to q_2$, then by definition

$$\begin{aligned}\bar{\phi}(F_1 + F_2) = \uparrow^{l_1 + l_2}((l_1 + x\#(l_2, p_1) + p_2) \cdot (\phi(f_1) + \phi(f_2))\\ \cdot (l_1 + x\#(q_1, l_2) + q_2)),\end{aligned}$$

and

$$\bar{\phi}(F_1) + \bar{\phi}(F_2) = \uparrow^{l_1}\phi(f_1) + \uparrow^{l_2}\phi(f_2).$$

Comparing the right sides of the above two equations it turns out that $\bar{\phi}(F_1 + F_2) = \bar{\phi}(F_1) + \bar{\phi}(F_2)$ is already a (synchronous) scheme identity, cf. (Bartha, 1987b, Claim C), i.e., it is provable from the identities SF. Thus, it is valid in the strong feedback theory A.

For a ranked alphabet Σ, let $T(\Sigma)$ denote the free algebraic theory generated by Σ.

COROLLARY 5.3. $F_s T(\Sigma) \cong \mathrm{Ft}(\Sigma)$.

Proof. Immediate by Proposition 3.6, since $T(\Sigma)$ is a subtheory of $\mathrm{Ft}(\Sigma)$ as well.

6. A Counterexample

One would expect that the free strong iteration theory generated by an algebraic theory T should always be the quotient of $F_s T$ by the congruence relation $\Omega(\mathrm{X})$. (Recall that X is the identity $\uparrow x = 1$.) This is not true, however, and here we give an example of an algebraic theory T for which $F_s T/\Omega(\mathrm{X})$ is not even a strong iteration theory.

Consider the ranked alphabet Σ consisting of two unary symbols τ_1, τ_2, and three ternary symbols σ_1, σ_2, and σ_3. Let θ denote the congruence relation of $T(\Sigma)$ induced by the equations

$$\sigma_1 \cdot (1 + \varepsilon) = \tau_1 \cdot \tau_2 + 0_1; \qquad \sigma_2 \cdot (1 + \varepsilon) = \tau_2 \cdot \tau_1 + 0_1;$$
$$\sigma_3 \cdot (1 + \varepsilon) = \tau_1 + 0_1.$$

Our example theory T will be $T(\Sigma)/\theta$. It is clear that T is closed, i.e., $T = \bar{T}$. Before going further, however, let us take a closer look at the structure of the iteration theory $F_s T/\Omega(\mathrm{X})$. For a morphism f in $T(\Sigma)$, let $f\theta$ denote the congruence class of θ containing f. Similarly, if $F = (l, f)$ is an $\mathrm{S}T(\Sigma)$-automaton, then let $F\theta$ denote the class $\mathscr{F} = (l, f\theta)$ in Aut_T.

Lemma 6.1. *Let $\mathscr{F} = (l, f\theta)$ and $\mathscr{G} = (m, g\theta)$ be $\mathrm{S}T$-automata $p \to q$. Then*

$$\mathscr{F} \sim \equiv \mathscr{G} \sim (\Omega(\mathrm{X}))$$

iff there exist $\mathrm{S}T(\Sigma)$-automata $H_i = (l_i, h_i)$, $i = 0, ..., n$, such that

- *$H_0 = (l, f)$, $H_n = (m, g)$;*
- *for every $j \in [n]$ one of the two conditions below is satisfied:*
  - (a) *$\mathscr{H}_{j-1} \to \mathscr{H}_j$ or $\mathscr{H}_{j-1} \leftarrow \mathscr{H}_j$, where $\mathscr{H}_i = H_i\theta$,*
  - (b) *$\uparrow^{l_{j-1}} h_{j-1} = \uparrow^{l_j} h_j$ holds in $\mathrm{It}(\Sigma)$.*

Proof. Observe that

$$\mathrm{It}(\Sigma) \cong \mathrm{Ft}(\Sigma)/\Omega(\mathrm{X}) \cong F_s(T(\Sigma))/\Omega(\mathrm{X}),$$

where the second isomorphism is determined by Eq. (3) above. This implies that it is enough to prove the lemma with condition (b) replaced by

(c) $H_{j-1} \equiv H_j$ $(\Omega(\mathrm{X}))$.

In this form the *if* part of the lemma is obvious. Now let us prove the *only if* part. Since

$$F_s T/\Omega(\mathrm{X}) \cong \mathrm{Aut}_T/(\sim \sqcup \Omega(\mathrm{X})),$$

there exist ST-automata $\mathscr{H}_0, \mathscr{H}_1, ..., \mathscr{H}_n$ such that

— $\mathscr{H}_0 = \mathscr{F}$, $\mathscr{H}_n = \mathscr{G}$;

— for every $j \in [n]$ one of (a), (d) below is satisfied:

(a) $\mathscr{H}_{j-1} \to \mathscr{H}_j$ or $\mathscr{H}_{j-1} \leftarrow \mathscr{H}_j$,

(d) $\mathscr{H}_{j-1} \equiv \mathscr{H}_j(\Omega(\mathrm{X}))$.

Since

$$\mathrm{Aut}_T/\Omega(\mathrm{X}) \cong \mathrm{Aut}_{T(\Sigma)}/(\theta \sqcup \Omega(\mathrm{X})),$$

condition (d) implies that there exist $ST(\Sigma)$-automata $L_0^j, ..., L_k^j$ such that

— $\mathscr{H}_{j-1} = L_0^j\theta$, $\mathscr{H}_j = L_k^j\theta$;

— for every $i \in [k]$,

$$L_{i-1}^j \equiv L_i^j(\theta) \qquad \text{or} \qquad L_{i-1}^j \equiv L_i^j(\Omega(\mathrm{X})).$$

Thus, condition (d) can be replaced by (c), as was to be proved.

Remark 6.2. In the above proof we made use twice of the fact that if A is a Q-algebra (Q is any type), ρ is a congruence relation of A, and E is a set of Q-identities, then

$$(A/\rho)/\Omega(E) \cong A/(\rho \sqcup \Omega_A(E)).$$

This statement is obvious by the second isomorphism theorem and by the fully invariant property of the congruence relation $\Omega(E)$. See, e.g., (Grätzer, 1968, 1979) for details.

Returning to our example, consider the ST-automaton $\mathscr{F} = (2, f\theta)$, where

$$f = (1 + x + x) \cdot (\langle \varepsilon \cdot \sigma_1 \cdot (1 + 0_1 + 2), \langle \varepsilon \cdot \sigma_2, \sigma_3 \rangle \cdot (0_1 + 3) \rangle),$$

see also Fig. 12. First we prove that

$$(\mathscr{F} \cdot \varepsilon) \sim \; = ((\varepsilon + 1) \cdot \mathscr{G}) \sim$$

holds in $F_s T/\Omega(\mathrm{X})$ for an appropriate $\mathscr{G}\colon 2 \to 1$. Indeed,

$$f \cdot (2 + \varepsilon) \equiv g'(\theta),$$

where

$$g' = (1 + x + x) \cdot (\varepsilon \cdot \tau_1 \cdot \tau_2 + \langle \varepsilon \cdot \tau_2 \cdot \tau_1, \tau_1 \rangle) + 0_1.$$

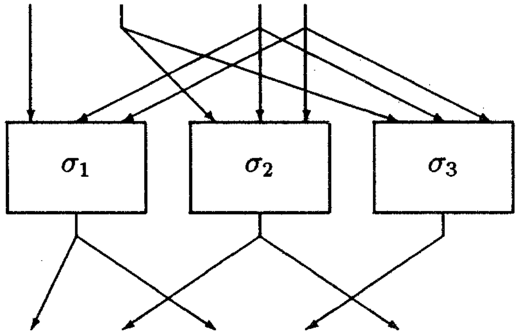

FIG. 12. The morphism $f: 5 \to 4$ of the counterexample.

Moreover, $\uparrow^2 g' = \uparrow g''$ holds in It(Σ) for the morphism

$$g'' = (1 + \varepsilon + 1) \cdot (\langle \varepsilon \cdot \tau_1 \cdot \tau_2, \tau_2 \rangle + 0_1).$$

Let $g = \langle \varepsilon \cdot \tau_1 \cdot \tau_2, \tau_2 \rangle + 0_1$, and put $\mathscr{G} = (1, g\theta)$. Applying Lemma 6.1 we obtain that

$$(\mathscr{F} \cdot \varepsilon) \sim \equiv ((\varepsilon + 1) \cdot \mathscr{G}) \sim (\Omega(\mathrm{X})).$$

If $F_s T/\Omega(\mathrm{X})$ were a strong iteration theory, then we should have $\uparrow^2 \mathscr{F} \sim = \uparrow \mathscr{G} \sim$ in it. We are going to prove, however, that this equation does not hold. A short computation shows that $\uparrow \mathscr{G} = (1, \varepsilon \cdot \tau_2 \cdot \tau_1)\theta$. But how can we handle $\uparrow^2 \mathscr{F} = (4, f\theta)$? Lemma 6.1 describes all the possibilities of representing $(4, f\theta)$ in a different form. Let $H = (l, h)$ be an arbitrary $\mathrm{S}T(\Sigma)$-automaton. We say that H is a σ-alternative of $(4, f)$ if $\uparrow^l h = \uparrow^4 f$ holds in It(Σ). To prove that in $F_s T/\Omega(\mathrm{X})$ $\uparrow^2 \mathscr{F} \sim \neq \uparrow \mathscr{G} \sim$, we need the following lemma.

LEMMA 6.3. *Let $H = (l, h)$ be a σ-alternative of* $(4, f)$, *and suppose that $K\theta \to H\theta$ or $H\theta \to K\theta$ holds for some* $\mathrm{S}T(\Sigma)$*-automaton $K = (k, t)$. Then K is still a σ-alternative of* $(4, f)$.

Clearly, $(1, \varepsilon \cdot \tau_2 \cdot \tau_1)$ is not a σ-alternative of $(4, f)$, hence by Lemmas 6.1 and 6.3 $\uparrow^2 \mathscr{F} = (4, f)\theta$ and $\uparrow \mathscr{G} = (1, \varepsilon \cdot \tau_2 \cdot \tau_1)\theta$ cannot be equivalent under $\Omega(\mathrm{X}) \sqcup \sim$.

Proof of Lemma 6.3. As we observed in Remark 3.3,

$$\to^i \subseteq \Omega(\mathrm{SF} \cup \mathrm{TH}) \subseteq \Omega(\mathrm{IT}).$$

It is therefore sufficient to consider simulations via surjective mappings in the lemma. (Recall that simulations have an epi–mono factorization by Proposition 4.10.) For the same reason we can assume that H is reduced.

Since $\uparrow^l h = \uparrow^4 f$, each component of h matches a subtree of the inifinite regular tree $f_\infty = \uparrow^4 f$ (up to only a finite depth, of course). The structure of f_∞ shows that h cannot be θ-equivalent to any Σ-tree containing occurrences of the symbols τ_1, τ_2.

Case a: $K\theta \to H\theta$. In this case

$$t \cdot \beta \equiv (\beta + 1) \cdot h(\theta)$$

for a surjective mapping $\beta: k \to l$. Then we must have $t \cdot \beta = (\beta + 1) \cdot h$, because otherwise h would be θ-equivalent to a tree containing $\tau_{1(2)}$ occurrences. It(Σ) being a strong iteration theory, this implies that K is a σ-alternative of $(4, f)$.

Case b: $H\theta \to K\theta$. Now we have

$$h \cdot \beta \equiv (\beta + 1) \cdot t(\theta)$$

for some surjective $\beta: l \to k$. Again, if β is an isomorphism or $h \cdot \beta = (\beta + 1) \cdot t$, then we are through. Suppose that $h \cdot \beta \neq (\beta + 1) \cdot t$ and β is not an isomorphism. In this case there exist distinct integers $i, j \in [l]$ such that $\beta(i) = \beta(j)$,

$$\pi^i \cdot h \equiv \pi^j \cdot h(\theta)$$

but $\pi^i \cdot h \neq \pi^j \cdot h$. Obviously, this property is inherited by some subtrees u_i and u_j of $\pi^i \cdot h$ and $\pi^j \cdot h$ such that their roots are already labelled differently. It is evident that the one label should be σ_1 and the other should be σ_3. Say $\mathrm{root}(u_i) = \sigma_1$ and $\mathrm{root}(u_j) = \sigma_3$. Since the leftmost branches of the subtrees u_i and u_j are $\sigma_1 \sigma_1 \cdots \sigma_{1-}$ and $\sigma_3 \sigma_2 \cdots \sigma_{2-}$, respectively, there must be some tree u the nodes of which are labelled by τ_1 and τ_2 only, and

$$u_i \equiv u \equiv u_j(\theta).$$

But this is impossible, since $u_i \equiv u(\theta)$ implies that $u = (\tau_1 \cdot \tau_2)^n \cdot \gamma_1$, and $u_j \equiv u(\theta)$ implies that $u = \tau_1 \cdot (\tau_2 \cdot \tau_1)^m \cdot \gamma_2$ for appropriate nonnegative integers n, m and injections $\gamma_1, \gamma_2: 1 \to k$.

7. Systolic Flownomials and Computer Architectures

Let us return to Ştefãnescu's and Căzănescu's Σ-flownomials over a theory T discussed in Section 4 to set up a further connection between iteration and feedback theories. The construction of the strong iteration theory $\mathrm{Fl}_{\Sigma,T}/\equiv_d$ (Ştefănescu, 1987), which parallels our F_s-construction, requires that T be a strong iteration theory as well. Observe that the whole construction of Şefănescu can be repeated almost word by word under the weaker assumption that T is only a strong *feedback* theory. Of course, the resulting theory $\mathrm{Fl}_{\Sigma,T}/\equiv_d$ will also be a strong feedback theory only. To make the necessary distinction, let $\mathrm{Sfl}_{\Sigma,T}$ denote the S-algebra (not detailed here) of *systolic* Σ-flownomials over T, which corresponds to $\mathrm{Fl}_{\Sigma,T}$ when T is a strong feedback theory. For a typical example, let $T = \mathrm{Pfn}_\nabla$ be the theory of partial ∇-trees; see (Bartha, 1987b, Definition 5.1). In other words, $\mathrm{Pfn}_\nabla \cong \mathrm{Ft}(\Sigma)$, with Σ being the void alphabet. On the analogy of (Ştefănescu, 1987, Corollary 5.3) it is clear that $\mathrm{Sfl}_{\Sigma,T}/\equiv_d$ is isomorphic to the sub-S-algebra of $\mathrm{Ft}(\Sigma)$ consisting of *systolic trees*. In terms of the representation of $\mathrm{Ft}(\Sigma)$ given in Section 3, an infinite tree t in $\mathrm{Ft}(\Sigma)$ is called systolic if the father of each Σ-labelled node in t is a ∇-node. Note that in this context the attribute "systolic" reflects faithfully the corresponding technical notion, because a systolic tree t can be considered as the unfolding of a synchronous Σ-scheme which is systolic in the sense of Leiserson and Saxe (1983). Interpreting the symbols of Σ as functional elements, t describes the behavior of such a generalized Moore automaton in which only the outputs corresponding to *ideal* components (i.e., those that are not variables) should be Moore-like. Speaking more technically, a machine of this kind is a Moore automaton with *broadcasting* components; see (Leiserson and Saxe, 1983).

What is the meaning of a systolic Σ-flownomial $F = \uparrow^n ((\sum_{i=1}^k \sigma_i + p) \cdot c)$: $p \to q$ in general? As we have seen in Section 4, the symbols σ_i are variables to be interpreted in the strong feedback theory T. For example, let $T = F_s(\mathrm{Pol}\, B)$, where $\mathrm{Pol}\, B$ is the theory of all polynomials of a suitable algebra B of logical gates over the set $\{0, 1\}$. Alternatively, we can choose $T = F_1^\infty(\mathrm{Pol}\, B_\perp)$ (see (Bartha, 1989)) if we prefer the step-by-step behavior of circuits, where $B_\perp$ is a suitable "pointed" extension of B over $\{0, 1\} \cup \perp$. Since each σ_i is interpreted as a finite state structural (and digital) Mealy automaton (i.e. a processor), the processor (system) corresponding to F will behave like a broadcasting Moore automaton. The flownomial F is therefore the model of a systolic computer architecture with q input and p output channels. In this architecture the morphism c represents the *fixed wiring* interconnecting the processors represented by the boxes σ_i. Concerning the processors, only their sort (i.e., the number of i/o ports) is fixed, otherwise their behavior is free. We can interpret this freedom by saying that they are *programmable* via an imaginary host machine and some i/o ports, which are invisible on this level of the model.

What is a *synchronous* flownomial? This problem needs a different treatment, because flownomials correspond to scheme expressions in normal forms, and the normal form of synchronous schemes, see (Bartha, 1987b, Definition 5.3), is more complex than that of (ordinary) flowchart shemes. The study of synchronous flownomials wil be the subject of a forthcoming paper.

ACKNOWLEDGMENTS

The author is thankful to all the members of the Programming Research Group in Oxford who helped his work at the Computing Laboratory, and also to G. Soros for sponsoring his research scholarship to Oxford. The paper benefited very much from the detailed comments of both referees. By their suggestions they gave him the most valuable help.

RECEIVED July 19, 1988; FINAL MANUSCRIPT RECEIVED July 27, 1990

REFERENCES

ARBIB, M. A., AND MANES, E. G. (1974a), Machines in a category: An expository introduction, *SIAM Rev.* **16**, 163–192.

ARBIB, M. A., AND MANES, E. G. (1974b), Foundations of system theory: Decomposable systems, *Automatica* **10**, 285–302.

ARBIB, M. A., AND MANES, E. G. (1980), Partially additive categories and flow-diagram semantics, *J. Algebra* **62**, 203–227.

ARNOLD, A., AND DAUCHET M. (1978, 1979), Théorie des magmoïdes, *RAIRO Inform. Théor. Appl.* **12**, 235–257 and **13**, 135–154.

BARENDREGT, H. P. (1984), "The Lambda Calculus: Its Syntax and Semantics," 2nd revised ed., North-Holland, Amsterdam.

BARTHA, M. (1987a), A finite axiomatization of flowchart schemes, *Acta Cybernet.* **8**, 203–217.

BARTHA, M. (1987b), An equational axiomatization of systolic systems, *Theoret. Comput. Sci.* **55**, 265–189.

BARTHA, M. (1988), Connections between feedback and iteration theories, submitted for publication.

BARTHA, M. (1989), Interpretations of synchronous flowchart schemes, *in* "Proceedings, 7th Conference on the Fundamentals of Computation Theory, Szeged," Lecture Notes in Computer Science, No. 380, pp. 25–34, Springer-Verlag, Berlin.

BARTHA, M. (1990), Foundations of a theory of synchronous systems, submitted for publication.

BENSON, D. B. (1975), The basic algebraic structures in categories of derivations, *Inform. Control* **28**, 1–29.

BLOOM, S. L., ELGOT, C. C., AND WRIGHT, J. B. (1980a), Solutions of the iteration equations and extensions of the scalar iteration operation, *SIAM J. Comput.* **9**, 25–45.

BLOOM, S. L., ELGOT, C. C., AND WRIGHT, J. B. (1980b), Vector iteration in pointed iterative theories, *SIAM J. Comput.* **9**, 525–540.

BLOOM, S. L., AND ÉSIK, Z. (1985), Axiomatizing schemes and their behaviors, *J. Comput. System Sci.* **31**, 375–393.

CĂZĂNESCU, V. E., AND ŞEFĂNESCU, G. (1988), "Feedback, Iteration and Repetition," Research Report 42, National Institute for Scientific and Technical Creation, Bucharest.

CULIK, K., II, AND FRIS, I. (1985), Topological transformations as a tool in the design of systolic networks, *Theoret. Comput. Sci.* **37**, 183–211.

ELGOT, C. C. (1975) Monadic computations and iterative algebraic theories, *in* "Logic Colloquium '73, Studies in Logic and the Foundations of Mathematics" (H. E. Rose and J. C. Shepherdson, Eds.), pp. 175–23, North-Holland, Amsterdam.

ELGOT, C. C., BLOOM, S. L., AND TINDELL, R. (1978), On the algebraic structure of rooted trees, *J. Comput. System Sci.* **16**, 228–242.

ELGOT, C. C., AND SHEPHERDSON, J. C. (1979), A semantically meaningful characterization of reducible flowchart schemes, *Theoret. Comput. Sci.* **8**, 325–357.

ELGOT, C. C., AND SHEPHERDSON, J. C. (1980), "An Equational Axiomatization of the Algebra of Reducible Flowchart Schemes," IBM Research Report RC 8221.

ÉSIK, Z. (1980), Identities in iterative and rational theories, *Comput. Linguist. Comput. Languages* **14**, 183–207.

ÉSIK, Z. (1982), On generalized iterative theories, *Comput. Linguist. Comput. Languages* **14**, 95–110.

ÉSIK, Z. (1988), Independence of the equational axioms for iteration theories, *J. Comput. System Sci.* **36**, 66–76.

GÉCSEG, F., AND PEÁK, I. (1972), "Algebraic Theory of Automata," Akadémiai Kiadó, Budapest.

GRÄTZER, G. (1968, 1979), "Universal Algebra," Springer-Verlag, Berlin.

HOTZ, G. (1965), Eine Algebraisierung des Syntheseproblems von Schaltkreisen, *Elektron. Inform. Kybernet.* **1**, 185, 205, 209–231.

HOTZ, G. (1966), Eindeutigkeit und Mehrdeutigkeit formaler Sprachen, *Elektron. Inform. Kybernet.* **2**, 235–247.

KALMAN, R., FALB, P., AND ARBIB, M. A. (1969), "Topics in Mathematical System Theory," McGraw–Hill, New York.

KELLY, G. M., AND STREET, R. (1974), Review of the elements of 2-categories, *in* "Category Seminar, Sidney 1972/73," pp. 76–103, Lecture Notes in Mathematics, No. 420, Springer-Verlag, Berlin.

LAWVERE, F. W. (1963), Functional semantics of algebraic theories, *Proc. Nat. Acad. Sci. U.S.A.* **50** (5), 869–872.

LEISERSON, C. E. (1982), "Area-efficient VLSI Computation," ACM–MIT Press Doctoral Dissertation Award Series 1, ACM–MIT, New York.

LEISERSON, C. E., AND SAXE, J. B. (1983), Optimizing synchronous systems, *J. VLSI Comput. Systems* **1**, 41–67.

MANES, E. G. (1976), "Algebraic Theories," Springer, Berlin.

MACLANE, S. (1971), "Categories for the Working Mathematician," Springer, Berlin.

PADULO, L., AND ARBIB, M. A. (1974), "System Theory," Saunders, Philadelphia.

ŞTEFĂNESCU, G. (1987), On flowchart theories. I. The deterministic case, *J. Comput. System Sci.* **35**, 163–191.

ŞTEFĂNESCU, G. (1986), "Feedback Theories (A Calculus for Isomorphism Classes of Flowchart Schemes)," Research Report 24, National Institute for Scientific and Technical Creation, Bucharest.

TROEGER, D. R. (1982), Metric iteration theories, *Fund. Inform.* **5**, 187–216.

WRIGHT, J. B., THATCHER, J. W. WAGNER, E. G., AND GOGUEN, J. A. (1976), Rational algebraic theories and fixed-point solutions, *in* "Proceedings, 17th IEEE Symposium on Foundations of Computer Science, Houston, Texas," pp. 147–158.

Printed in Belgium